Conference Record

Volume 1 of 2
(BOUND IN ONE VOLUME)

PAPERS PRESENTED
OCTOBER 31-NOVEMBER 2, 1988
PACIFIC GROVE, CALIFORNIA

Twenty-Second Asilomar Conference on Signals, Systems & Computers

EDITED BY
Ray R. Chen
San Jose State University
San Jose, CA

SPONSORED BY
NAVAL POSTGRADUATE SCHOOL
Monterey, California
SAN JOSE STATE UNIVERSITY
San Jose, California
IN COOPERATION WITH
IEEE Computer Society

IEEE Computer Society Order Number 871
Library of Congress Number 89-050387
IEEE Catalog Number 88-CH2660-9
ISBN 0-929029-15-1

88CH2835-7

THE COMPUTER SOCIETY OF THE IEEE
THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.
MAPLE PRESS

88

Pour le titre du livre
voir en seconde page

p. 843 - 847

Detection and Location of Seismic Events in Real Time

J. Frederking†, N. Magotra‡

† Dept. of EECE, University of New Mexico, Albuquerque, NM 87131

‡ Dept. of EECE, University of New Mexico, Albuquerque, NM 87131 and Sandia National Laboratories, Albuquerque, NM 87185

ABSTRACT

This paper presents a real time seismic event detection and source location (RSEDSL) algorithm using single station (three component) data for seismic analysis. RSEDSL detects and locates an event through parameters derived from the data covariance matrix in a manner analogous to beam steering using data from an array of single component stations. The algorithm has been implemented on a TMS32020 real time digital signal processing (DSP) system and tested on several data sets. The algorithm has also been implemented on a Sun 2/170 mainframe. Comparison of the results obtained from the TMS with those obtained from the Sun indicate that seismic events can be accurately detected and located in real time.

1. Introduction

Seismic analysis using single station (three component) seismic data has important applications as a means of verifying a nuclear test ban treaty [1]. The objective of this paper is to show that seismic events can be accurately detected and located in real time. To accomplish this objective RSEDSL has been implemented on a TMS32020 [2,3] DSP system and tested on several seismic events. The results were then compared to those obtained from an equivalent implementation on a Sun 2/170 mainframe. The results from the the real time implementation on the TMS32020 were comparable to those obtained from the Sun.

RSEDSL is designed to detect and locate regional seismic events using short period seismic data (sampled at 40 Hz) as input. A regional seismic event is defined to be one in which the angle subtended at the center of the earth by an arc connecting the receiver and source (epicenter) is less than 20°. This angle (Δ) corresponds to a surface distance of 2224 km [4]. The algorithm has been extensively tested using data obtained from a regional seismic test network (RSTN) originally designed and built by Sandia National Laboratories (Albuquerque, NM 87185). Each station in this network senses earth motion along three orthogonal axes North (N), East (E), and vertical (Z). This data was originally sampled at 40 Hz and later lowpass filtered at 3.5 Hz and resampled at 8 samples per second (sps). As currently implemented, RSEDSL uses only the N and E data channels for event detection. Once an event is detected, the instantaneous source azimuth ($\hat{\phi}(m)$) is determined by estimating the polarization direction of the initial compressional phase (P_n). The source azimuth is shown as $\hat{\phi}$ in Figure 1. The Z-channel is used to resolve an inherent 180° ambiguity in $\hat{\phi}(m)$. The distance to the source is then determined from the time difference between the P_n phase and a later phase (L_g). RSEDSL is also capable of detecting intermediate (S_n) phases.

2. The algorithm

RSEDSL begins by estimating the input data covariance matrix $\hat{C}(m)$ of the N and E input channels. $\hat{C}(m)$ is defined as

$$\hat{C}(m) = \begin{bmatrix} \hat{\sigma}_n^2(m) & \hat{\sigma}_{ne}(m) \\ \hat{\sigma}_{ne}(m) & \hat{\sigma}_e^2(m) \end{bmatrix} \quad (1)$$

where m is the time index, $\hat{\sigma}_n^2(m)$ is the N channel (n(m)) variance, $\hat{\sigma}_e^2(m)$ is the E channel (e(m)) variance, and $\hat{\sigma}_{ne}(m)$ is the N, E channel data crosscorrelation. Note that each of the data channels is assumed to be zero mean. The following recursive relations are used to compute each element of $\hat{C}(m)$ [1]

$$\hat{\sigma}_n^2(m) = \beta[\hat{\sigma}_n^2(m-1) - n^2(m)] + n^2(m) \quad (2a)$$

$$\hat{\sigma}_e^2(m) = \beta[\hat{\sigma}_e^2(m-1) - e^2(m)] + e^2(m) \quad (2b)$$

$$\hat{\sigma}_{ne}(m) = \beta[\hat{\sigma}_{ne}^2(m-1) - n(m)e(m)] + n(m)e(m) \quad (2c)$$

where β is a smoothing parameter with a value between 0 and 1. For the implementation discussed in this paper β has a value of 0.98.

22ACSSC-12/88/0843 $1.00

Event detection is accomplished through the use of a detection parameter derived from the maximum eigenvalue ($\hat{\lambda}_M(m)$) of $\hat{C}(m)$. $\hat{\lambda}_M(m)$ is obtained by solving the characteristic equation of $\hat{C}(m)$ as follows

$$\hat{\lambda}_M(m) = \frac{\hat{\sigma}_n^2(m) + \hat{\sigma}_s^2(m) + \sqrt{(\hat{\sigma}_n^2(m) - \hat{\sigma}_s^2(m))^2 + 4\hat{\sigma}_{ns}^2(m)}}{2} \quad (3)$$

A smoothed delayed value of $\hat{\lambda}_M(m)$ is then computed recursively as

$$\hat{\lambda}_S(m) = \beta[\hat{\lambda}_S(m-1) - \hat{\lambda}_M(m-D)] + \hat{\lambda}_M(m-D) \quad (4)$$

In equation (4) β is a smoothing constant equal to 0.98 and D is a delay parameter equal to 200 samples. The detection parameter $\hat{\zeta}_s(m)$ is defined as

$$\hat{\zeta}_s(m) = \hat{\lambda}_M(m) - k_s\hat{\lambda}_S(m) > 0 \ ? \quad (5)$$

RSEDSL declares an event when $\hat{\zeta}_s(m)$ exceeds zero. Once an event has been declared, a new event flag is set and detection is disabled until $\hat{\lambda}_M(m)$ returns to its pre-event noise level. The pre-event noise level is determined by the value of $\hat{\lambda}_S(m)$ at the time the event is declared.

The onset time of an event is determined by backing up a maximum of 10 seconds prior to the detection of the event and computing a second detection parameter. This detection parameter ($\hat{\zeta}_l(m)$) is computed as in equation (5) above except that the scalar multiplier k_s is replaced by a smaller value k_l. In the current implementation k_s is 3.5 and k_l is 1.5.

After RSEDSL has detected the onset of an event, it then obtains an estimate of the direction of travel of the seismic signal in the NE-plane as measured from the N axis. The technique that is used is analogous to beamforming with a seismic array. The instantaneous azimuth is computed from the eigenvector corresponding to $\hat{\lambda}_M(m)$ and is given by

$$\hat{\phi}(m) = \tan^{-1}\left[\frac{\hat{\lambda}_M(m) - \hat{\sigma}_n^2(m)}{\hat{\sigma}_{ns}(m)}\right] \quad (6)$$

An estimate of $\hat{\phi}(m)$ is obtained by averaging equation (6) over a period of 2.0 sec. starting 1.0 sec. after the onset time of the event [1].

As previously mentioned the distance to the source of an event is determined from the time difference between the P_n and L_g phases. RSEDSL is capable of detecting and differentiating these two phases. Standard tables are then used to obtain an estimate of the source-to-receiver distance [4]. This task has been left to the analyst in the current implementation.

The algorithm has been tuned to detect events with body wave magnitudes (m_b) in the range 3.0 to 6.0. Tests using a Fortran implementation of the algorithm on a Sun 2/170 computer have shown that events having a source-to-receiver distance of 1300 km and m_b in the range 4.0 to 6.0 could be consistently detected and located to within a radius of approximately 70 km of the source [1]. The results obtained from the implementation on the Sun 2/170 were used as benchmarks with which to measure the performance of the real time implementation on the TMS32020.

3. Implementation

RSEDSL was implemented on a 20 MHz TMS32020 based DSP system supplied by Pacific Microcircuits Ltd. [6]. The system was designed to fit in an expansion slot of an IBM PC-XT. Data was transferred between the PC and Sun 2/170 via an Ethernet link. In addition to the TMS32020's 544 words of on-chip RAM, the board is equipped with 16K of high speed external RAM. The board also has 16 bit A/D and D/A ports with analog to digital conversion time of 17 μs. Note that for the implementation discussed in this paper the input data was 'sampled' from the on-board external RAM and not the A/D.

The most serious problems encountered during the implementation of the algorithm were truncation and round-off errors resulting from the conversion of 32 bit products to 16 bits. Initially, these errors were large enough to cause RSEDSL to yield unsatisfactory results, especially in events with a weak P_n phase [7]. Reducing the effects of these errors required the use of 32 bit arithmetic wherever possible and altering the form of the equations used to estimate the covariance matrix. These equations (2a-c) were written in the following form in the original algorithm

$$y(m) = \beta y(m-1) + (1-\beta)x^2(m) \quad (7)$$

Implementing this equation in Fortran on the Sun 2/170 presented no problems since the floating point data format provided sufficient dynamic range. However, this is not a very efficient form when implemented on the fixed point TMS32020 and results in serious underflow problems for small input values.

Whenever the TMS32020 multiplies two 16 bit numbers, the resulting 32 bit product is saved in a product register whose contents can be directly added to the contents of the 32 bit accumulator. This property was used to reduce underflow errors in estimating the covariance matrix elements by imple-

menting equation (7) in the following form

$$y(m) = \beta[y(m-1) - x^2(m)] + x^2(m) \quad (8)$$

One can see in this equation that if the output is saved in 32 bit format, then the term in brackets can be evaluated as a 32 bit number before being converted to 16 bits for multiplication with β.

Another property of the TMS32020 that helped facilitate the use of 32 bit arithmetic in RSEDSL was the normalization instruction. This instruction was used to convert 32 bit numbers to 16 bit numbers for multiplication by shifting left a number of times until a one was in the highest bit position. The number of left shifts was saved in an auxilliary register to be used for denormalization after multiplication.

A special approximation subroutine was used to implement the square root term in equation (3). This term was of the general form

$$R = \sqrt{I^2 + Q^2} \quad (9)$$

Many methods exist for obtaining a close approximation of R if I and Q are known [8]. For the implementation discussed in this paper I and Q were determined via the following relations

$$I = |\delta_n^2(m) - \delta_e^2(m)| \quad (10a)$$

$$Q = |2\delta_{ne}(m)| \quad (10b)$$

Equation (9) was implemented by using the following formulas [8].

$$\hat{R} = |I| + \frac{|Q|}{2} \qquad |I| > |Q| \quad (11a)$$

$$\hat{R} = |Q| + \frac{|I|}{2} \qquad |Q| > |I| \quad (11b)$$

Equations (11a,b) were very simple to implement on the TMS32020. Since division by two only requires a right shift of one bit, $\hat{R}$ can be determined by a shift and an addition. This approach proved to be better than other schemes tried [7].

Computation of $\hat{\phi}(m)$ in equation (6) also required the use of an approximation subroutine to implement the arctangent function. This subroutine used one of two approximation techniques depending upon the magnitude of the argument (x):

$0.0 \le x < 0.3$: $\arctan(x) = x$

$0.3 \le x < 1.0$: use look-up table. 250 entries

resolution = .0028 rad.

Prior to calling the subroutine the magnitudes of the numerator and denominator in equation (6) were compared. If the numerator was smaller, then x was less than one and an approximation was determined by the appropriate technique as indicated above. If the numerator was larger than the denominator, then the two were swapped prior to dividing and a flag was set. An approximation was then made by one of the above techniques and the following relation was used to account for the swap

$$\alpha = 90^0 - \beta \quad (12)$$

Figure 2 shows a plot of the absolute error between the output of this subroutine and that of a standard C library arctan function evaluated on the Sun 2/170. Note that this error is less than $.5^0$ (.00873 rad.). This approximation scheme was found to be more accurate and simpler to implement than other schemes tried [7].

The entire program (RSEDSL) was stored in the high speed external memory while the arctan look-up table, derived parameters, and all temporary variables were stored in the on-chip RAM.

4. Results

RSEDSL took 220 μs to process each input sample vector (n(m), e(m), z(m)) if no event was detected. This figure increased to 340 μs if an event was detected. This time includes determination of onset time and azimuth averaging and is independent of which phase was detected (P_n, S_n, or L_g). These figures indicate that RSEDSL is capable of processing data from a network of several stations even at the original sampling rate of 40 sps.

Figures 3, 4, and 5 show plots of some of the outputs obtained on the Sun 2/170 and TMS32020. The input used to obtain these results corresponds to a nuclear test at the Nevada test site observed at an RSTN station in Albuquerque, NM. The source-to-receiver distance in this case is approximately 800 km. The event had a body wave magnitude of 4.2 [4]. Figure 3 shows the input Z-channel, maximum eigenvalue, smoothed delayed maximum eigenvalue, and azimuth estimate from the TMS32020 for the entire event. Figure 4 shows the same outputs expanded about the P_n phase but with the outputs from the Sun 2/170 (dotted line) superimposed. Figure 5 also shows the same outputs as Figure 3 but expanded about the L_g phase and with the outputs from the Sun 2/170 (dotted line) superimposed. Note that the maximum eigenvalue plots are identical, while the azimuth plots differ before the onset of the event but are nearly identical once the event starts. Table 1 shows a comparison of detection times and azimuth estimates for each phase of the event as determined on the Sun 2/170 and TMS32020. Note that detection times on the TMS32020

are within 0.125 seconds and the azimuth estimates are within three degrees for the important P_n and L_g phases. RSEDSL was tested on two additional seismic events (nuclear tests) and yielded similar results [7].

5. Conclusions

RSEDSL has been tested on data corresponding to actual underground nuclear tests and acheived results comparable to those obtained on a Sun 2/170 computer. The execution time was short enough to allow RSEDSL to process multiple stations in real time. Finally, RSEDSL needs to be tested on a larger number of events with varying magnitudes and source-to-receiver paths.

References

[1] Magotra N., Ahmed N., Chael E., "Seismic Event Detection and Source Location Using Single-station (Three-Component) Data", *Bull. Seism. Soc. Am.*, pp. 958-971, June 1987.

[2] *TMS32020 User's Guide*, Texas Instruments Inc., 1985.

[3] *Digital Signal Processing Applications with the TMS320 Family*, Texas Instruments Inc., 1985.

[4] Simon, R. B., *Earthquake Interpretations - A Manual for Reading Seismograms*, William Kaufman, Inc., 1981.

[5] Breding, D. R., "Data User's Guide for the Regional Seismic Test Network (RSTN)", Rept. SAND 82-2953. 87pp., Sandia National Lab, Albuquerque, NM 87185.

[6] *TMS32020 Development System User Manual*, Pacific Microcircuits Ltd., Blaine, WA 98230, 1986.

[7] Frederking, J., "Real-Time Seismic Event Detection and Source Location Using A TMS32020 Digital Signal Processor", Master's thesis, Department of Electrical and Computer Engineering, University of New Mexico, 1988.

[8] Filip, A. E., "A Baker's Dozen Magnitude Approximations and Their Detection Statistics", *IEEE Trans. Aerospace and Electronics Systems*, Jan. 1976.

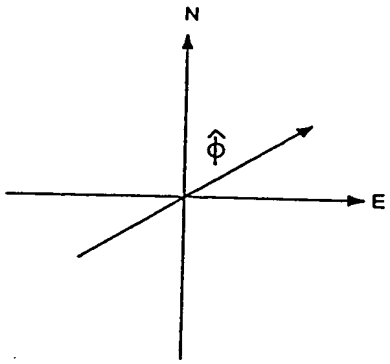

Figure 1. Diagram of NE azimuth

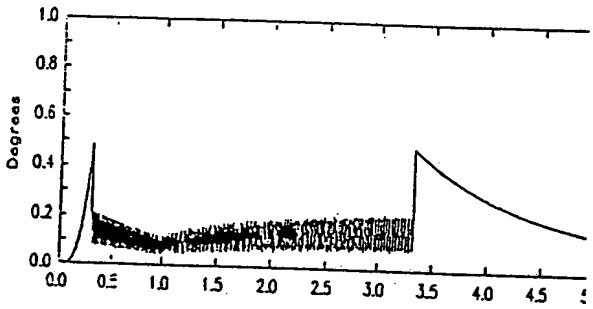

Figure 2. Error plot of Arctan (x)

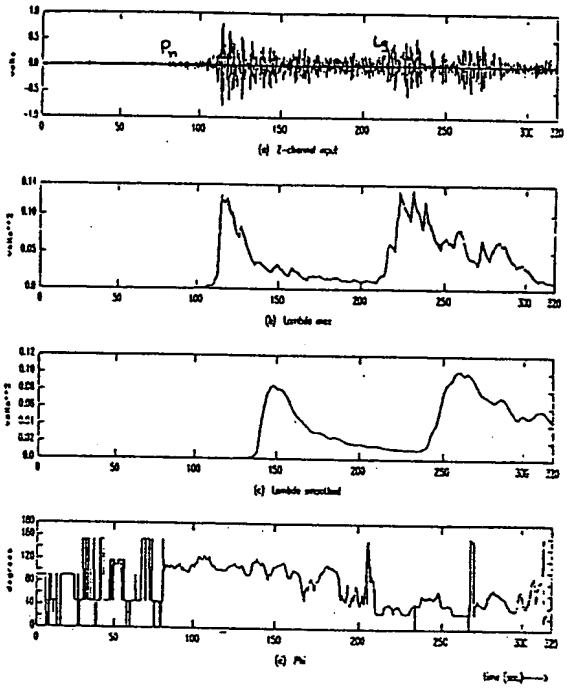

Figure 3. Plots of outputs from TMS32020

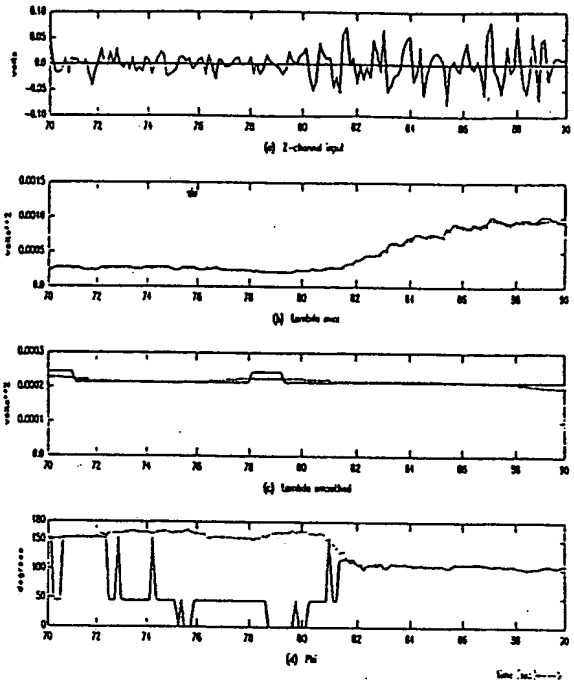

Figure 4. Comparison of outputs from TMS (solid line) with Sun for Pn phase

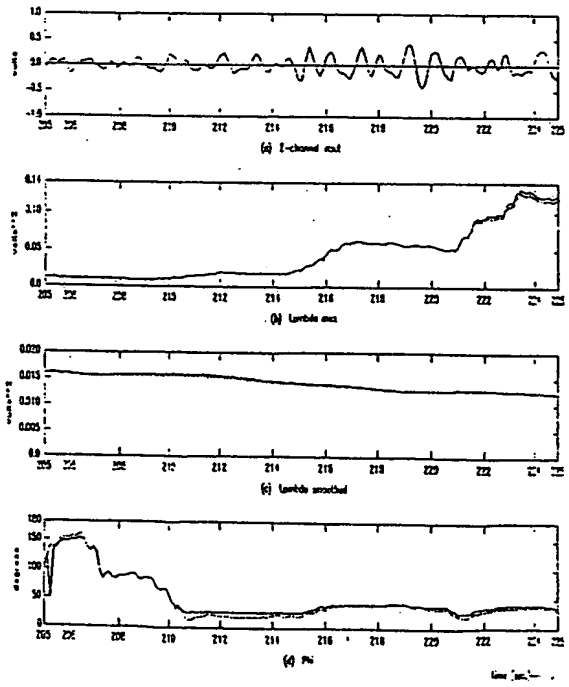

Figure 5. Comparison of outputs from TMS (solid line) with Sun for Lg phase

	TMS32020	SUN 2/170
P_n PHASE	85.5	85.375
ONSET TIME	81.5	81.625
NE AZM	107.51	104.49
NEXT PHASE	115.375	115.5
ONSET TIME	105.5	105.625
NE AZM	123.72	100.96
L_g PHASE	216.0	216.125
ONSET TIME	214.5	214.625
NE AZM	37.91	40.04

Time in seconds. NE AZM in degrees.

Table 1. Comparison of output.